



## COURS DE D'ELPHI

### SÉRIE N°03

### LES PROPRIÉTÉS D'UN OBJET

#### OBJECTIF PÉDAGOGIQUE :

À l'issue de cette série, le stagiaire doit être capable d'effectuer les modifications sur les différentes propriétés qui existent.

#### PLAN DE LA LEÇON :

##### **I. DÉFINITION D'UNE PROPRIÉTÉ**

##### **II. LES PROPRIÉTÉS LES PLUS IMPORTANTES**

##### **III. MODIFIER LA VALEUR D'UNE PROPRIÉTÉ**

1. Modification en mode création

2. Modification en mode exécution

#### **CONCLUSION**

#### **EXERCICES D'APPLICATION**

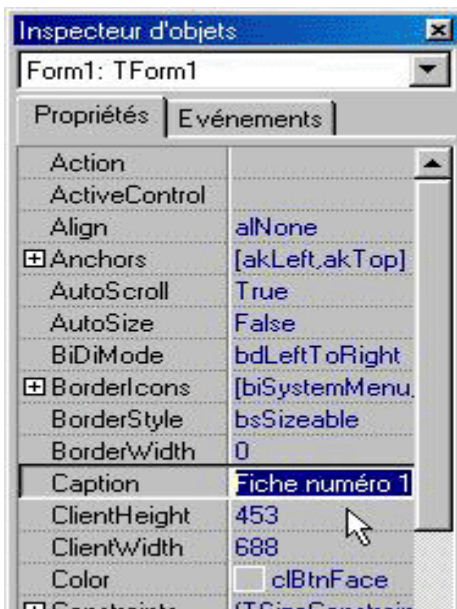
#### **SOLUTIONS**

#### **LES PROPRIÉTÉS D'UN OBJET**

## I- DÉFINITION D'UNE PROPRIÉTÉ :

Comme on l'a déjà vu au niveau de la **SÉRIE N°01« L'environnement Delphi 7 »**, l'Inspecteur d'objet est composé de deux onglets Propriétés et Événements (qu'on verra de plus près dans la prochaine série). Dans cette série, on va s'intéresser aux différentes propriétés des composants de Delphi.

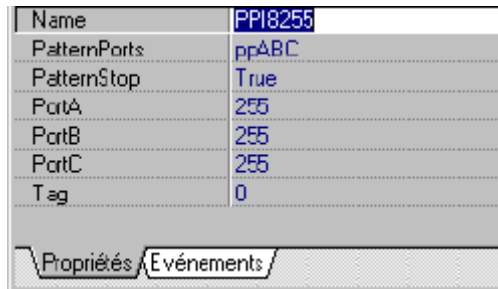
L'onglet **Propriétés** dans **L'Inspecteur d'Objet** affiche une liste de propriétés (caractéristiques) pour chaque composant qu'on insère au niveau de la fiche, si la fiche est vide donc elle ne contient aucun composant, dans ce cas l'inspecteur d'objet affiche tout simplement les propriétés de la fiche elle-même.



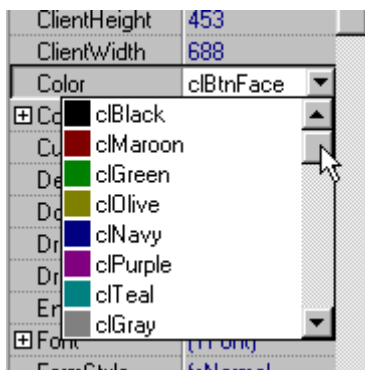
## II-LES PROPRIÉTÉS LES PLUS IMPORTANTES :

Parmi les propriétés importantes et qu'on utilise souvent :

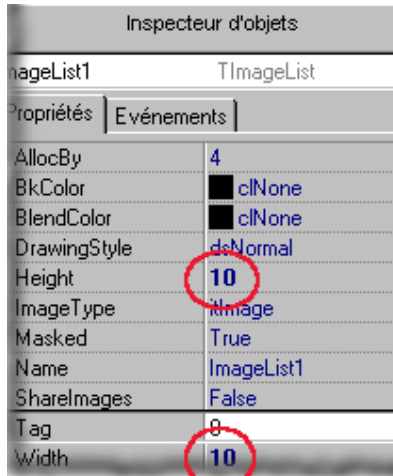
- ❖ Name ; *Name* est une propriété commune à tous les composants, elle a pour valeur le nom du composant ou de la fiche. Par défaut l'inspecteur d'objet attribut des noms qu'il numérote automatiquement.



- ❖ Color Type enum, a pour valeur la couleur du fond de la fiche ou du composant.



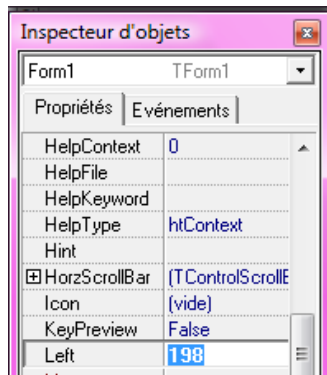
- ❖ **Width** Type entier, nous permet de déterminer la largeur de notre composant.



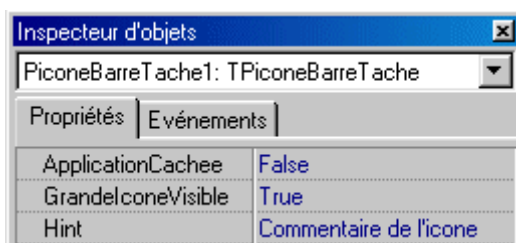
- ❖ **Top** Type entier, on utilise la propriété *Top* pour déterminer l'emplacement du bord supérieur d'un composant ou pour le repositionner différemment.



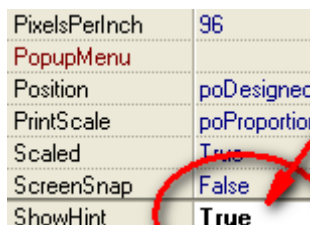
- ❖ **Left** Type entier, on utilise la propriété *Left* pour déterminer l'emplacement du bord gauche d'un composant ou pour le repositionner différemment.



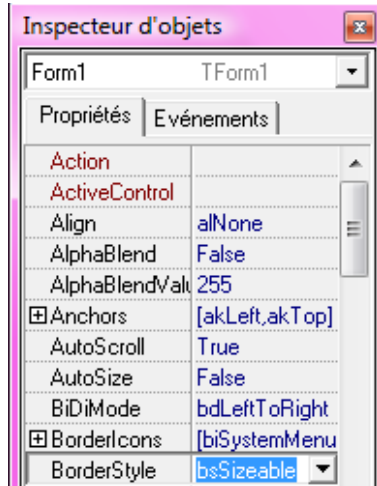
- ❖ **Hint** Type String, on Utilise la propriété *Hint* pour indiquer un texte d'aide qu'on affiche à un endroit spécifique, par exemple sur la barre d'état.



- ❖ **Showhint** Type booléen, la propriété Showhint permet d'afficher le texte qu'on a inséré au niveau de *Hint* et pour cela il suffit de mettre sa valeur à **True**.



- ❖ **Border style** Type en um, nous permet de définir le style de bordure de notre composant.



### III- MODIFIER LA VALEUR D'UNE PROPRIÉTÉ :

#### 1- Modification en mode création :

Dans ce mode, la modification agit sur le composant dès la conception (d'où le principe de programmation visuel, on peut s'apercevoir de chaque modification au fur et à mesure qu'on effectue la conception).

#### 2- Modification en mode exécution :

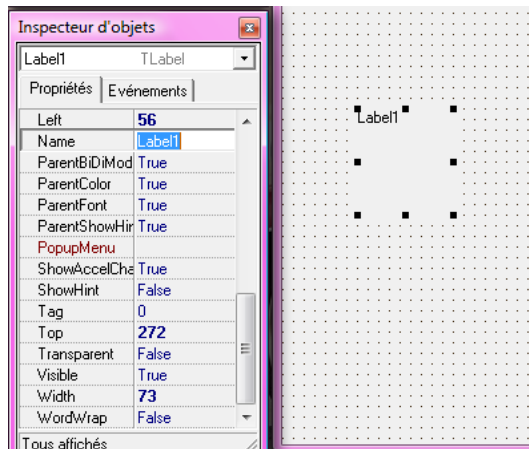
Dans ce mode en revanche, la modification ne prend effet qu'à l'exécution de l'application. C'est le cas de la propriété **Visible** lorsqu'on lui affecte la valeur **False**, on remarque qu'il n'y a aucun changement. Cette valeur a pour effet de rendre invisible le composant à l'exécution.

### CONCLUSION :

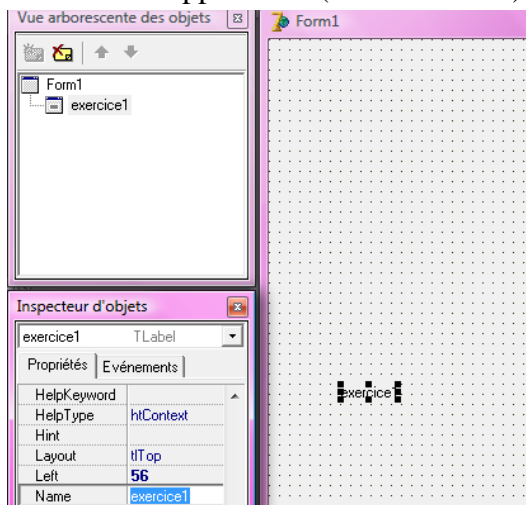
À travers la leçon suivante on a pu comprendre la notion de **propriété** dans Delphi, on a vu les plus utilisées et on a constaté que leur modification pouvait se faire de deux modes différents. Ces notions vont être mieux comprises à travers l'exercice d'application qui va suivre.

# EXERCICES D'APPLICATION

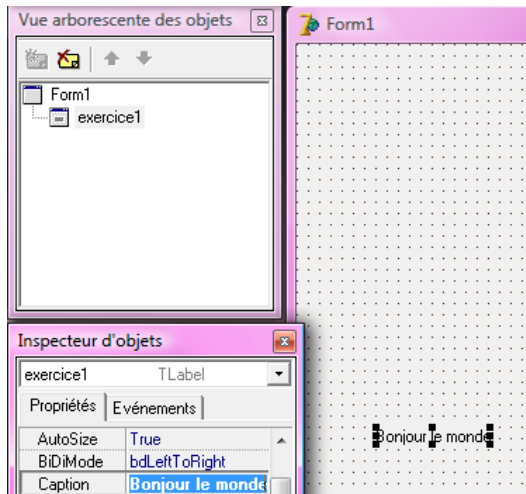
## EXERCICE N° 1 :



1- Au niveau de l'inspecteur d'objet, visualiser la propriété *Name* par défaut c'est « Label 1 », vous pouvez le changer en introduisant votre nouvelle appellation (Ex : exercice1).



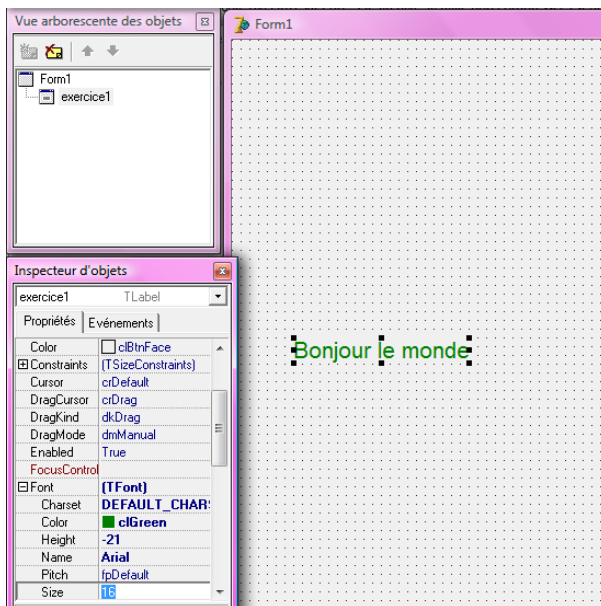
- 2- Sélectionnez la propriété *Caption*, changez ça valeur en mettant à la place de la valeur par défaut, la valeur « Bonjour le monde », vous allez remarquer que le texte change directement sur le composant *Label1*.



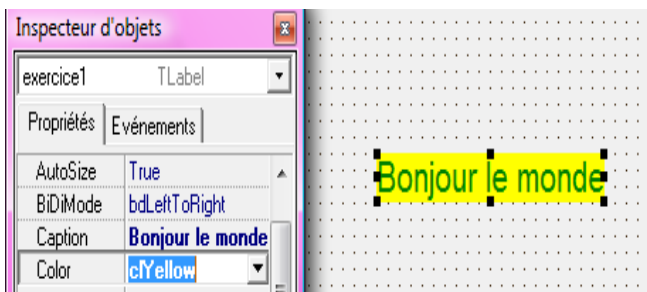
- 3- La propriété *Font* contient plusieurs sous-propriétés, afin de les modifier il suffit de cliquer sur le signe (+) qui précède la propriété *Font* :

- Mettez la valeur « CLGreen » au niveau de la propriété Color (couleur des caractères).
- Mettez la valeur « Arial » pour la propriété Name (nom de police).
- Mettez la valeur 16 pour la propriété Size (taille des caractères).





- 4- Pour modifier la couleur de fond du label, il suffit de modifier la propriété Color qui se situe après la propriété Caption et lui affecter la valeur « CLYellow ».



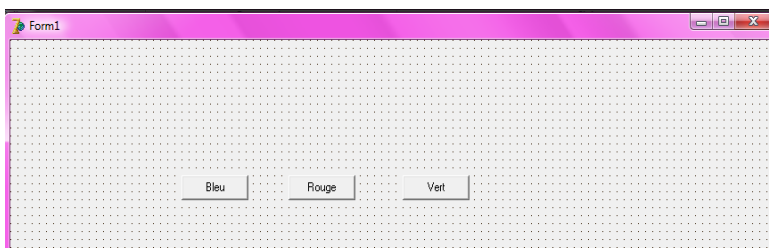
## SOLUTION EXERCICEN° 1 :

- 1- Insérer les composants suivants de la palette Standard en cliquant



sur le composant et en modifiant par la suite la propriété **Caption** selon le tableau suivant :

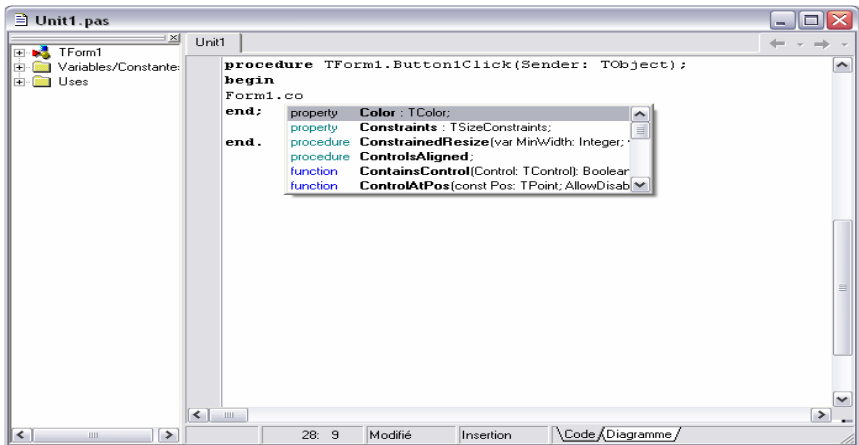
| Composant | Propriété | Valeur |
|-----------|-----------|--------|
| Boutton1  | Caption   | Bleu   |
| Boutton2  | Caption   | Rouge  |
| Boutton3  | Caption   | Vert   |



- 2- Double cliquez sur le bouton1 (Bleu) et écrire dans la procédure : **Form1.Color:=CLBlue;**

### Remarque :


Si vous écrivez : Form1. Delphi, grâce à son assistant de saisie semi-automatique, vous propose la liste des membres (propriétés et méthodes) de la classe saisie (Form1), et on peut soit choisir dans liste la propriété ou méthode voulue, ou taper les premières lettres de l'élément voulu (exemple : co pour écrire color) et lorsqu'il sera sélectionné dans la liste on clique sur Entrée pour que Delphi l'écrit dans l'instruction.



3- taper F12 pour revenir à la fiche et double cliquez sur le bouton2 (Rouge) et écrire dans la procédure **:Form1.color:=CLRed;**

4- taper F12 pour revenir à la fiche et double cliquez sur le bouton3 (Vert) et écrire dans la procédure :

**Form1.color:=CLGreen;**

5- Lancer l'exécution par la touche F9, ou par le bouton ; Et vérifier le bon fonctionnement de la fiche, en cliquant sur chacun des boutons la couleur du fond va changer selon l'intitulé du bouton.

## **LEÇON 2 : LES ÉVÈNEMENTS D'UN OBJET**

**OBJECTIF PÉDAGOGIQUE** : À l'issue de cette leçon, le stagiaire doit être capable de gérer un d'un programme.

### **PLAN DE LA LEÇON :**

**I-DÉFINITION D'UN ÉVÈNEMENT**

**II-LES ÉVÈNEMENTS LES PLUS IMPORTANTS**

**CONCLUSION**

**EXERCICES D'APPLICATION**

**SOLUTIONS**

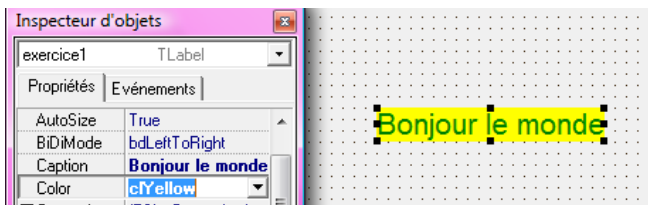
## LEÇON 2 : LES ÉVÈNEMENTS D'UN OBJET

### I- DÉFINITION D'UN ÉVÈNEMENT

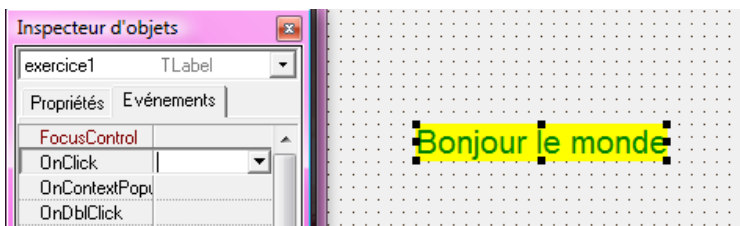
Un évènement permet de déclencher une procédure, chaque objet possède des évènements qui peuvent le gérer. Le rôle d'un évènement sera plus clair à travers l'exemple suivant :

Pour voir le sens d'un évènement, revenons à l'exemple d'application précédent (**Leçon 1 : Les propriétés d'un objet/exercice 1**) :

À la fin de l'exercice, nous avons obtenu le résultat suivant :

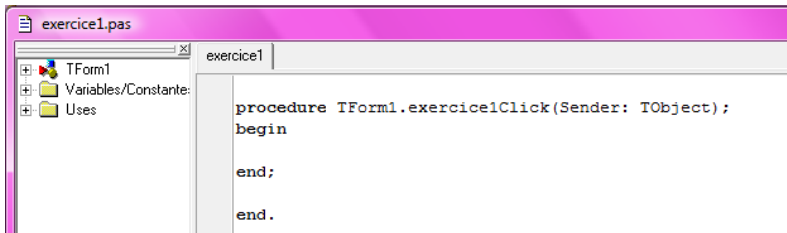


- Cliquez une fois sur le label « Bonjour le monde » pour le sélectionner.
- Remettez la propriété *Visible* à **True**.
- Cliquez sur l'onglet évènement pour voir apparaître les évènements relatifs au composant Label.



- Sélectionnez l'évènement *On Clicket* double cliquez sur la zone de saisie ou bien tous simplement double cliquez sur le composant lui-même :

La fenêtre du code apparaît à l'écran et une procédure relative à l'évènement et crée automatiquement (elle apparaît vers la fin du code) :



Il ne reste qu'à écrire les instructions relatives à l'action qu'on veut avoir.

L'évènement *OnClick* signifie qu'on exécute une action lorsqu'on clique sur le composant *Label1*, comme exemple nous allons essayer de créer l'évènement suivant :

Lorsqu'on clique sur le label « **Bonjour le monde** », on doit avoir le texte « évènement d'un objet » qui s'affiche à sa place avec un fond bleu, pour cela rajoutons l'instruction suivante :

Procédure TForm1.exercice1Click (Sender : TObject) ;

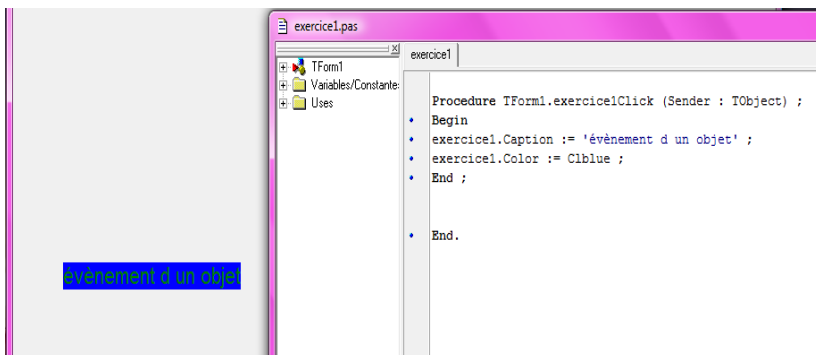
### Begin

exercice1.Caption := 'évènement d'un objet' ;

exercice1.Color := Clblue ;

### End ;

Exécutez le programme, puis cliquez sur 'Bonjour le monde ' et constatez le résultat.



## II-LES ÉVÉNEMENTS LES PLUS IMPORTANTS :

Ci-dessous un tableau qui regroupe les évènements les plus utilisés dans Delphi avec leur signification :

| Evènement    | Définition   |
|--------------|--|
| OnDbl click  | L'action se déroule lorsqu'on <b>double-clic</b> sur l'élément en question.  |
| On Mous Down | L'action se déroule lorsqu'on clique sur le bouton de la souris au-dessous d'un paragraphe.  |
| On Mous UP   | L'action se déroule lorsque l'utilisateur relâche le bouton de la souris au-dessous d'un élément.  |
| On Mous Move | L'action se déroule lorsque le curseur de la souris est en mouvement au-dessous d'un élément.  |
| OnKeyPress   | L'action se déroule lorsque l'utilisateur clique sur une touche du clavier.  |
| On KeyUp     | L'action se déroule lorsque l'utilisateur relâche une touche du clavier.   |
| On Key Down  | L'action se déroule lorsque l'utilisateur clique sur une touche du clavier.  |
| On create    | Utilisez OnCreate pour effectuer des actions spécifiques quand une fiche est créée et invoquée par le constructeur de TCustomForm.   |
| On activate  | Utilisez l'événement OnActivate pour effectuer des actions particulières quand la fiche reçoit la focalisation. Une fiche devient active quand elle obtient la focalisation, par exemple quand l'utilisateur clique dans la fiche. |
| On close     | L'évènement est utilisé lorsqu'on souhaite produire une action à la fermeture de l'application.  |

## CONCLUSION :

À travers cette leçon nous avons pu voir le rôle d'un évènement dans une procédure Delphi ainsi que les différents évènements qui existent. Ces informations vont se renforcer à travers les exercices d'application qui vont suivre.

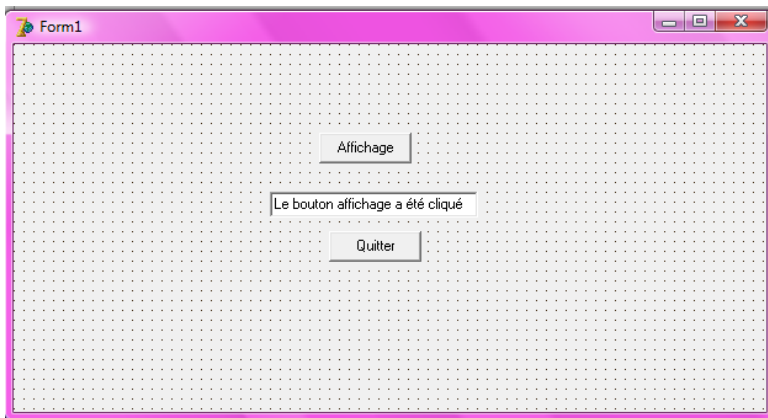
## EXERCICES D'APPLICATION :

### EXERCICE N° 1 :

Concevez l'application qui effectue les opérations suivantes :

- 1- Sur une même fenêtre, lorsque vous cliquez sur un bouton, le message « Affichage du message » s'affiche dans une zone de texte.
- 2- Lorsqu'on clique sur un autre bouton sur la fenêtre, cette dernière se ferme et l'application se termine.

Vous devez obtenir la forme suivante :



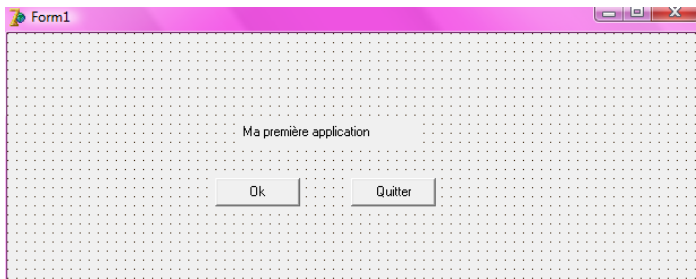
### EXERCICE N° 2 :

Concevez l'application suivante :

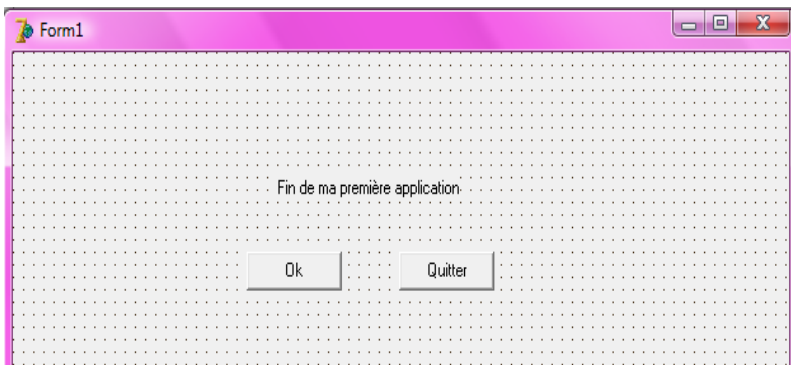
- 1- Sur une fenêtre insérez la zone d'affichage 'Ma première application'.
- 2- Lorsqu'on clique sur un bouton le label change en 'Fin de ma première application'.
- 3- Insérez un autre bouton afin de quitter l'application.



Vous devez obtenir la forme suivante :



Lorsqu'on clique sur le bouton **Ok** la fenêtre devient comme suit :



## SOLUTIONS

### EXERCICE N°1 :

Afin de concevoir l'application, suivez les étapes suivantes :

- Exécutez Delphi pour créer une nouvelle application
- Placez un bouton de commande sur la forme **Forme1**, ce bouton a le nom de **Bouton1**.
- Dans l'onglet Propriétés de l'inspecteur d'objet, affectez à sa propriété *Caption* la valeur 'Affichage'.
- Placez une zone de saisie *Edit* sur la fenêtre **form1**, ce contrôle a pour nom **Edit1**.
- Sélectionnez le contrôle Edit, et dans l'onglet Propriétés de l'inspecteur d'objet, effacez la valeur de la propriété *Text*.
- Insérez un deuxième composant *Button* et attribuez la valeur 'Quitter' à sa propriété *Caption*. Ce bouton a pour nom *Button2*.
- Dimensionnez la fenêtre pour qu'elle soit plus présentable.

À présent il faut ajouter le code qui correspond à chaque bouton :

- Double cliquez sur le bouton *Affichage*, Delphi va créer la déclaration de la procédure correspondante à l'événement *OnClick* du bouton, la procédure est comme suit :  
**Procédure TForm1.Button1Click (Sender :TObject) ;**  
**Begin**  
**Edit1.Text := 'Le bouton affichage a été cliqué' ;**  
**End ;**
- Double cliquez sur le bouton *Quitter*, dans la procédure de l'événement *OnClick* entrez l'instruction suivante :  
**Procédure TForm1.Button2Click(Sender :TObject) ;**  
**Begin**  
**Close ;**  
**End ;**

- Dans le menu **Fichier**, sélectionnez **Tout Enregistrer** et enregistrez votre projet.
- Complétez et exécutez le projet.

## EXERCICE N° 2 :

Afin de réaliser l'exercice, suivez les étapes suivantes :

- Exécutez Delphi pour créer une nouvelle application.
- Dans l'onglet **Propriétés** de l'inspecteur d'objet, définissez la propriété *Caption* de cette fenêtre et lui attribuer la valeur « Ma première application ».
- Définissez la propriété *Height* de la fenêtre et lui attribuer la valeur '250'.
- Définissez la propriété *Width* de la fenêtre et lui attribuer la valeur '450'.
- Dans la palette des composants, choisissez l'onglet **Standard**, placez un bouton de commande *Button* sur la fenêtre *Form1*. Puis lui affecter à la propriété *Caption* la valeur 'Ok'.
- Cliquez sur le signe '+' de la propriété *Font*, les autres propriétés s'affichent.
- Définissez la propriété *Size* et lui attribuer la valeur '14'.
- Définissez la propriété *Name* de la police en choisissant dans la liste la valeur 'Arial'.
- Insérez un deuxième composant *Button* et attribuez la valeur 'Quitter' dans *Caption*.
- Définissez ses propriétés de police (*Size*, *Name*) comme pour le bouton *Ok*.
- Placez une zone d'affichage *Label* sur la fenêtre *Form1*.
- Définissez la propriété *Caption* de ce bouton en lui attribuant la valeur 'Ma première application'.
- Redimensionner la fenêtre pour qu'elle soit mieux présentée.

À présent l'interface est prête, reste à ajouter le code :

- Double cliquez sur le bouton **Ok** afin d'introduire le code de la procédure qui va permettre de remplacer le texte existant par 'Fin de ma première application' :

**Procedure TForm1.Button1Click (Sender :TObject) ;**

**Begin**

**Label1.Caption := 'FIN de ma première application' ;**

**End ;**

- Revenez à l'affiche en cliquant sur F12, puis double cliquez sur le bouton **Quitter** et ajouter la procédure suivante :

**Procedure TForm1.Button2Click (Sender :TObject) ;**

**Begin**

**Close ;**

**End ;**

- Dans le menu **Fichier**, sélectionnez **Tout Enregistrer** et enregistrer votre projet.
- Une boîte de dialogue s'affiche, sélectionnez votre répertoire de travail, donnez un nom à l'unit1 puis enregistrez. Donnez aussi un nom au projet puis enregistrer, l'unité sera enregistrée sous l'extension '.drp'.
- Enfin compilez et exécutez le projet, en choisissant **Exécuter** dans le menu **Exécuter** ou en appuyant sur **F9**.